

HT-C302
ARINC-429 总线
CPCI 插卡
技术使用说明书

天津市英贝特航天科技有限公司

1 概述

CPCI ARINC429 总线接口板(代号 HT-C302)是 3U CPCI 专用 ARINC429 接口板。有六个 ARINC429 接收通道和三个 ARINC429 发送通道,ARINC429 总线数据传输速率可选为 100Kbps、50Kbps、48Kbps 和 12.5Kbps，在每一设定总线数据传输速率点,允许总线数据传输速率变化范围±10%。386EX 微处理器处理 ARINC429 的数据通信，通过板内一个双口 RAM 实现主机与后端电路交换信息,具有 CPCI 高稳定性、高可靠性和可选的热插拔功能。

2 主要技术指标

★ 满足 ARINC429 总线通讯标准
★ 33M 32bit CompactPCI
★ PCI9030 PCI 总线接口(热插拔功能可选)
★ 集成 ARINC429 六路接收通道和三路发送通道。ARINC429 总线数据传输速率跳线可选为 100Kbps、50Kbps、48Kbps 和 12.5Kbps
★ 使用 HS1-3282-8 ARINC429 控制器和 HS1-3182-8 ARINC429 总线驱动器，在设定的每个总线数据传输速率点允许数据传输速率变化范围±10%
★ 内嵌 Intel 386EX 微处理器局部处理 429 数据通信
★ 128K 字节数据存储器，128K 字节程序存储器
★ 内嵌 16K 字节双口存储器
★ 板尺寸：3U 160mm×100mm
★ 工作温度：-10～＋70℃
★ 提供 WINDOWS、Wince、Linux、VxWorks 系统程序支持

3 硬件使用说明

3.1 硬件组成

HT-C302 硬件组成框图如图一所示，386EX 实时监视并处理 ARINC429 信息；128K 字节程序存储器存放实时处理程序；128K 字节数据存储器存放实时处理程序的数据资源；HS1-3282-8ARINC429 控制器实现 ARINC429 接口控制；PCI9030 为 CPCI 总线接口, LTC1643L 负责热插拔控制；16K 字节双口存储器介于 PCI9030 的 LOCAL BUS 与 386EX 的 separate BUS 之间实现后端电路与主机交换命令和信息（数据）；HT-C302 主要元件位置如图二所示（图中方焊盘为元件的第一管脚）。

图一、硬件组成框图

图二、HT-C302 板主要元件位置图

3.2 跳接针使用说明

3.2.1 ARINC429 收发速率选择

HT-C302 可以通过配置 E8 来选择 ARINC429 总线传输速率。

表一　ARINC429 传输速率选择表

总线传输速率	E8/1-2	E8/3-4
100Kbps	OFF	OFF
50Kbps	OFF	ON
48Kbps	ON	OFF
12.5Kbps	ON	ON

3.2.2 前后 I/O 输出跳接针使用说明

HT-C302 通过 J4、J5、J6 选择连接 ARINC429 接收和发送通道到前面板 DB25 或后面板 CPCI J2，如下：

连接到前面板：需要分别连接 J4、J5、J6 的（脚 2、脚 1）、（脚 5、脚 4）、（脚 8、脚 7）、（脚 11、脚 10）（脚 14、脚 13）、（脚 17、脚 16）。

连接到后面板：需要分别连接 J4、J5、J6 的（脚 2、脚 3）、（脚 5、脚 6）、（脚 8、脚 9）、（脚 11、脚 12）（脚 14、脚 15）、（脚 17、脚 18）

3.2.3 其它跳接针使用说明

HT-C302 的其它跳接针为维修测试专用，用户禁止使用，以免破坏系统设置。

3.3 对外接口信号定义

说明：

a. 429 通讯时可以只使用差分的两根信号线。但是如果将两台设备的地线连接会对通讯的可靠性有好处。

b. HT-C302 与外接口采用前面板插座 DB25 芯孔或后面板 CPCI J2。

表二　接口信号定义

芯号	信号名称	说　明	芯号	信号名称	说　明
1	R1+	接收 1＋	2	R1-	接收 1－
3	R2+	接收 2＋	4	R2-	接收 2－
5	R3+	接收 3＋	6	R3-	接收 3－
7	R4+	接收 4＋	8	R4-	接收 4－
9	S1+	发送 1＋	10	S1-	发送 1－
11	S2+	发送 2＋	12	S2-	发送 2－
13		未用	14	GND	
15	S3+	发送 3＋	16	S3-	发送 3－
17	R5+	接收 5＋	18	R5-	接收 5－
19	R6+	接收 6＋	20	R6-	接收 6－
21		未用	22		未用
23		未用	24		未用
25		未用			未用

3.4 CPCI 连接器

HT-C302 的 J2（CPCI_J1）连接到标准 32bit CPCI 背板。
HT-C302 的 J3（CPCI_J1）为 I/O 连接器，连接到 CPCI 背板，如下

通道 1
接收 R1+_B---J3.C4
接收 R1-_B---J3.D4
接收 R2+_B---J3.C5
接收 R2-_B---J3.D5
发送 S1+_B---J3.C10
发送 S1-_B---J3.D10

通道 2
接收 R3+_B---J3.C6
接收 R3-_B---J3.D6
接收 R4+_B---J3.C7
接收 R4-_B---J3.D7
发送 S2+_B---J3.C11
发送 S2-_B---J3.D11

通道 3
接收 R5+_B---J3.C8
接收 R5-_B---J3.D8
接收 R6+_B---J3.C9
接收 R6-_B---J3.D9
发送 S2+_B---J3.C12
发送 S2-_B---J3.D12

4 软件使用说明

4.1 ARINC-429 数据格式

4.1.1 ARINC-429 32 位格式

ARINC-429 32 位格式是指数据传输线上的格式。

ARINC 位	功能	说明
1-8	Label	地址标号
9-10	SDI	信息源/目标，表明该物理量从何处来
11	LSB	数据低位
12-27	Data	数据位。当用不满时，高位对齐，低位填零
28	MSB	数据高位
29	Sign	符号位。0 表示正、北、东，1 表示负、南、西
30-31	SSM	00：故障告警 01：非计算数据

		10：功能测试 11：正常操作
32	Parity Status	校验位。0：奇校验，1：偶校验

4.1.2 ARINC-429 编程格式

ARINC-429 编程格式是指对控制器芯片 HS-3282 访问的格式，也是对双口 RAM 访问的格式，在本说明书中称 ARINC 字，一个 ARINC 字为 32 位，共四个字节。

编程位	功能	ARINC 位
31	Sign	29
30	MSB	28
29-14	Data	27-12
13	LSB	11
12,11	SDI 或 Data	10,9
10,9	SSM Status	31,30
8	Parity Status	32
7-0	Label	1-8

说明：在某些设备中，符号位用 SSM Status 表示，00 为正，11 为负。31-11 均为数据位（共 21 位，也有只使用低 15 位的），如下图所示：

位	31 11	10，9	8	7 0
说明	(MSB) 数据 (LSB)	符号	校验	Label

使用本板之前，请跟与本板相接的设备方制定具体的协议。

4.2 双口 RAM 协议

HT-C302 板上配置有 16K 字节的双口存储器，用于局部 CPU 与主机交换信息，实现六路接收、三路发送、自检等功能。双口存储器的地址为 0000～3FFF，其约定如下：

序号	名称		偏移地址	字节数	说明	初始值	相对于主机 R／W
1	接收通道缓冲区	通道 1	0-3FF	1024	每个 ARINC 字占四个字节，以先低后高方式存放，存放顺序与其 LABLE 大小相一致。	0	R
		通道 2	400-7FF				
		通道 3	800-BFF				
		通道 4	C00-FFF				
		通道 5	1000-13FF				
		通道 6	1400-17FF				
2	发送通道缓冲区	通道 1 单发	1800-1BFF	1024	每个 ARINC 占四个字节，以先低后高方式存放。	0	W
		通道 1 循环发	1C00-1FFF				
		通道 2 单发	2000-23FF				
		通道 2 循环发	2400-27FF				
		通道 3 单发	2800-2BFF				
		通道 3 循环发	2C00-2FFF				
3	发送个	通道 1 单发	3000	1	最大为 255。等于 0 时不发送。	0	W
		通道 1 循环发	3001				
		通道 2 单发	3002				

序号	名　称		偏移地址	字节数	说　　明	初始值	相对于主机R／W
	数	通道2循环发	3003				
		通道3单发	3004				
		通道3循环发	3005				
4	接收通道产生中断的LABLE值	通道1	3100	1	接收的ARINC字的LABLE与其相等时向主机发一个中断并设中断标志。	0	W
		通道2	3101				
		通道3	3102				
		通道4	3103				
		通道5	3104				
		通道6	3105				
5	接收通道的中断标志	通道1	3106	1	为55H时表示是该通道向主机发了一个中断，主机响应中断查询该单元后要将该单元清0	0	R
		通道2	3107				
		通道3	3108				
		通道4	3109				
		通道5	310A				
		通道6	310B				
6	接收个数	通道1	3200－3201	2	实时记录接收通道接收的ARINC字的个数，每接收到一个ARINC字，该值增一	0	R
		通道2	3202－3203				
		通道3	3204－3205				
		通道4	3206－3207				
		通道5	3208－3209				
		通道6	320A－320B				
7	命令控制字		3FF0	1	C0 设置发送通道1的发送数据间隔	0	W
					C1 设置发送通道2的发送数据间隔		
					C2 设置发送通道3的发送数据间隔		
					C3 自检命令		
					C4 设置接收控制命令		
					C5 取消接收控制命令		
					C6 设置校验命令		
					C7 设置发送器速率命令		
					C8 设置接收器速率命令		
					C9 设置数据长度命令		
8	命令参数缓冲区		3FF1－3FF2	2	在写入各命令前，要先在该缓冲区中写入命令参数	0	W
9	自检标志		3FF3	1	自检标志，详见4.6的说明。	0	R
10	主机撤消中断		3FFC－3FFD	2	以字方式读该单元使中断撤消	0	R

注：a. LABLE（地址标号）为每个ARINC字的低八位。
b. 双口存储器支持字和字节两种操作。

4.3 信号灯的分配和使用

4.3.1 关于信号灯的说明

HT-C302板采用双口存储器的方式实现与主机的数据交换。当板上的局部CPU和主机同时访问双口存储器时，板上的忙状态仲裁逻辑会保证让一方先访问存储器，另一方处于等待状态，直

到先访问的一方访问结束时为止，这样就保证了每次读/写存储器的正确性。但是，要保证读写一组数据的完整性，仅有忙状态仲裁逻辑还不够。例如：一个 ARINC 字由 32 位组成，板上的存储器为 16 位或 8 位访问，局部 CPU 要分两次 16 位操作才能读写一个 ARINC 字；同样，主机也要分两次 16 位操作才能读写一个 ARINC 字。如果对同一个 ARINC 字在双口存储器两端同时发生读和写操作，就有可能造成读写的 ARINC 字高 16 位和低 16 位不是一个完整的值。信号灯正是为了保证一组数据的读写完整性而引入的。要访问存储器的一方先申请信号灯，申请到信号灯后，再对存储器进行读写操作，操作结束后释放信号灯。

信号灯可以使用双口存储器的某个单元来实现，但这种方式存在不可靠的因素。HT-C302 板上的信号灯是由硬件实现的，可以保证使用信号灯的双方可靠地工作。

4.3.2 信号灯的分配

HT-C302 板共有八个信号灯，以存储器方式访问，地址由段地址加偏移地址组成，段地址与双口存储器的段地址一致，偏移地址共占用 16K 字节空间，即 4000～7FFF，但只使用其中的 16 字节单元，如下表所示：

名　称	地址	字节数	对应的存储区
信号灯 1	4000－4001	2	接收通道 1 缓冲区 0-3FF
信号灯 2	4002－4003	2	接收通道 2 缓冲区 400-7FF
信号灯 3	4004－4005	2	接收通道 3 缓冲区 800-BFF
信号灯 4	4006－4007	2	接收通道 4 缓冲区 C00-FFF
信号灯 5	4008－4009	2	接收通道 5 缓冲区 1000-13FF
信号灯 6	400A－400B	2	接收通道 6 缓冲区 1400-17FF
信号灯 7	400C－400D	2	发送通道缓冲区 1800-2FFF
信号灯 8	400E－400F	2	备用

信号灯访问必须用字操作。

信号灯只是在访问对应存储区时使用，访问其它单元时不要使用。

局部 CPU 在上电时已将局部 CPU 端的八个信号灯释放；**主机在初始化时应将八个信号灯释放一遍，以保证初始状态**。在没有获得信号灯以前，连续申请信号灯等效于一次申请信号灯；在获得信号灯以后，连续释放信号灯等效于一次释放信号灯。申请和释放的操作见 4.3.3。

4.3.3 信号灯的使用

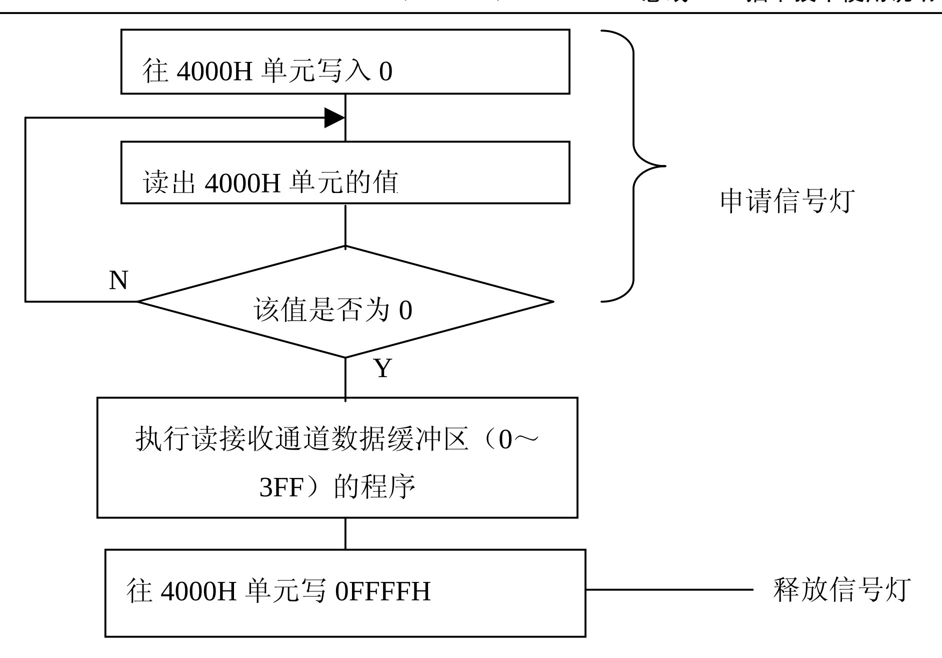
对信号灯的操作有申请信号灯和释放信号灯两种。当要访问双口存储器时，需要申请信号灯；当访问完后，要释放信号灯。**占用信号灯的程序要尽可能的短**，最好只执行读写双口存储器的操作。

申请信号灯：往相应信号灯单元写入字 0，然后读该地址，如读出为 0，则申请上了信号灯；否则继续读该地址，直到为 0 时为止。

释放信号灯：往相应信号灯单元上写入字 0FFFFH。

4.3.4 信号灯的例子

以读接收通道 1 缓冲区为例：



4.4 中断的编程说明

HT-C302 对双口存储器的 0x3ffc 单元以字方式写 0xffff 产生主机中断，主机以字方式读 0x3ffc 单元撤销中断;主机对双口存储器的 0x3ffe 单元以字方式写 0xffff 可产生 HT-C302 中断，HT-C302 以字方式读 0x3ffe 单元撤销中断。注意：0x3ffc 和 0x3ffe 两个单元不可再做其他用途。主机中断接总线的哪一级中断由跳接针选择，见 3.2.2 的说明。

4.5 接收工作方式

4.5.1 接收方式的工作原理

HT-C302 板的六路接收是不受控制一直打开的。也就是说，当任何一路接收通道有数据过来时，HT-C302 板就接收该数据并存放在约定的接收缓冲区中，主机任何时候从双口存储器中读取的都是最新收到的数据。

数据的存放顺序是按照其 ARINC 字的 LABLE 值大小依次存放的,每个 ARINC 字都对应 4 个字节。例如，LABLE 为 03 的 ARINC 字在接收通道 2 中存放在 40C-40F 中。主机可以通过 LABLE 值乘以 4 加上缓冲区首址来间接寻址，方便地实现读取所需要的数据。

4.5.2 接收中断及标志的产生

由于 HT-C302 板是无条件接收并不断刷新接收缓冲区的，因此约定了接收中断的产生条件，即 4.2 中所述的 3100、3101、3102、3103、3104、3105 单元。当接收到的 ARINC 字的 LABLE 值与上述单元的值相等时，就向主机发一个中断，同时在相应的中断标志上置 55H。每个通道都有一个中断标志单元。当主机使用中断方式时，通过查询中断标志单元可以确定是哪个通道产生的中断；主机也可以不使用中断方式，直接查询相应的中断标志单元。

由于接收事件是随时发生的，有时一个中断可能对应几个通道的接收，因此，主机响应中断后，应将六个中断标志单元全部查询一遍。

中断的撤消由主机以字方式读一次 3FFC 单元来执行。每次进入中断后，主机必须执行撤消中断的操作，同时将对应的中断标志单元清零。

4.5.3 接收计数

接收个数（3200～320B）用于记录每个接收通道的接收次数。对于一个特定的通道，HT-C302 板只要收到一个数据，就将对应计数单元增一。主机可以通过读取该单元的值判断接收的数据是

否被更新。

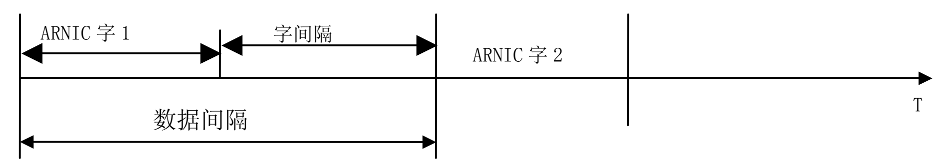
4.5.4 其它注意事项

由于 HT-C302 板是无条件接收的，所以主机无法辨别出那些 ARINC 字是新发来的，那些 ARINC 字是以前发的。如果主机需要辨别出哪些数据被更新，可以在每次接收之后将相应的接收通道缓冲区清零来处理。

4.6 发送工作方式

4.6.1 发送数据间隔

发送数据间隔指第一个 ARNIC 字到第二个 ARNIC 字的间隔时间。如下图所示：



数据间隔由用户编程控制，其方法为：

先在 3FF1－3FF2 单元写入要修改的数据间隔值，单位为 μs，然后再在 3FF0 单元写控制命令字 C0（C1、C2），则可以修改发送通道的发送数据间隔。

例如，要把发送通道 2 的数据间隔改为 1250μs，则先在 3FF1-3FF2 单元写 E2 04，然后在 3FF0 单元写 C1，就完成了发送通道 2 的数据间隔的设置。主机执行完 C1 命令后，将 3FF0 单元清零。

上电时，局部 CPU 将三个发送通道的数据间隔编程为 1250μs。

4.6.2 发送帧间隔

按照 ARINC429 的总线协议，字间隔最小为四个码位。通讯速率为 100K、48K、12.5K 或 6K 时，对应最小字间隔和最小数据间隔如下表所示：

通讯速率	最小字间隔	最小数据间隔
100K	40μs	360μs
48K	84μs	750μs
12.5K	320μs	2880μs
6K	672μs	6000μs

一旦编程的数据间隔小于最小数据间隔制，将导致控制器芯片工作不正常。这时，即便将时间再编程为正常值，也不能恢复控制器芯片的正常工作。使控制器芯片正常工作的方法是硬件复位或关电后重新加电。

一组 ARNIC 字为一帧。帧间隔指一帧数据的间隔时间。帧间隔因为发送方式的不同而不同：主机控制发送方式（以下简称单发）时，帧间隔由主机控制。**注意，主机控制的帧间隔时间要大于数据间隔乘以数据总数的积，否则会导致非正常发送**；自动循环发送方式时，帧间隔与数据间隔和发送数据总数有关。它们的关系为：

单发方式：

帧间隔>数据间隔*数据总数

自动循环发送方式：

帧间隔＝数据间隔*数据总数

4.6.3 单发方式发送控制

单发方式由发送个数（3000，3002，3004）和发送缓冲区（1800－1BFF，2000－23FF，2800－2BFF）共同控制。发送个数决定发送的一帧总数，不能为 0；发送缓冲区确定发送的数据，主

机在每发送一帧数据前，应先检查相应单发通道的发送个数缓冲区是否为 0，如不为 0，说明上一帧数据没有发完，则应等待，直到其值为 0 时再发下一帧数据。局部 CPU 执行完一帧数据的发送后，将发送个数清零。

4.6.4 自动循环方式发送控制

自动循环发送方式由发送个数（3001，3003，3005）和发送缓冲区（1C00－1FFF，2400－27FF，2C00－2FFF）共同控制。发送个数决定发送的一帧总数，不能为 0；发送缓冲区确定发送的数据，局部 CPU 按照发送个数和帧间隔自动循环发送。当主机将发送个数写为 0 时，停止发送。

单发方式比自动循环发送方式优先级高，当一个通道的两种方式都存在时，只执行单发方式。

4.6.5 发送控制实例

实例 1：单发方式

以发送通道 1 为例，先查询发送个数 3000 单元，当其值为 0 时，在发送通道缓冲区 1800-1817 写下列数据：

BF 00 00 10 D8 06 00 20 D9 00 00 30 B6 06 00 40 B7 00 00 50 B4 06 00 60

再在发送个数 3000 单元写 06，则发送通道 1 以数据间隔 1250μs（缺省值）**发送一次** LABLE 为 BF,D8,D9,B6,B7,B4 的六个数据。

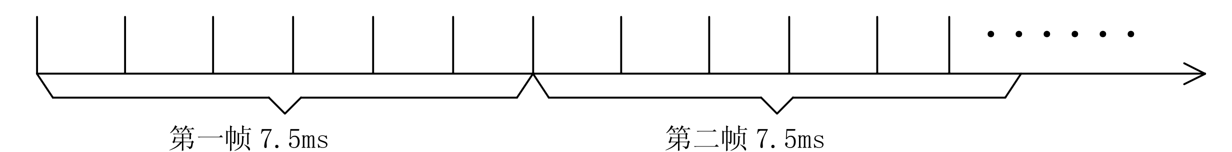
写发送通道缓冲区 1800-1817 时，要按 4.3 的说明使用信号灯。

实例 2：自动循环发送方式

以发送通道 1 为例，在发送通道缓冲区 1C00-1C17 写下列数据：

BF 00 00 10 D8 06 00 20 D9 00 00 30 B6 06 00 40 B7 00 00 50 B4 06 00 60

再在发送个数 3001 单元写 06，则发送通道 1 以数据间隔 1250μs（缺省值）、帧间隔 7.5ms 循环发送 LABLE 为 BF,D8,D9,B6,B7,B4 的六个数据。如下图所示：



4.7 自检工作方式

HT-C302 板可以通过两种方式进行自检，一种是在上电初始化时，自动执行一次自检；另一种是主机在 3FF0 单元处写入自检命令字 C3，启动一次自检。局部 CPU 执行自检命令时，将对六个接收通道进行检测，如果全部接收通道自检正确，就将 HT-C302 板把“429OK”信号置上高电平；只要有一个接收通道检测不正常，就将“429OK”信号置为低电平。自检结果还同时写入自检标志 3FF3 单元。

自检标志共八位，定义如下：

位 0：为 0 时表示第一个接收通道自检正确；为 1 时表示错误。
位 1：为 0 时表示第二个接收通道自检正确；为 1 时表示错误。
位 2：为 0 时表示第三个接收通道自检正确；为 1 时表示错误。
位 3：为 0 时表示第四个接收通道自检正确；为 1 时表示错误。
位 4：为 0 时表示第五个接收通道自检正确；为 1 时表示错误。
位 5：为 0 时表示第六个接收通道自检正确；为 1 时表示错误。
位 6～位 7：没有定义

主机启动自检后，应延时 20μs 再读自检标志单元。

4.8 控制命令

4.8.1 设置接收控制命令

C4 命令是设置接收控制命令。在执行该命令前,应先在 3FF1 和 3FF2 单元处填上命令参数，3FF1 单元处填写的是接收通道号,范围从 1 到 6; 3FF2 单元处填写的是接收控制码，它只有最后两位有效。当执行完命令后，每收到一个 ARINC 字，就判断所收到 ARINC 字的第 10、9 两位与接收控制码的第 1、0 两位是否分别相等，若相等，则接收该 ARINC 字，若有不等的，则不收该 ARINC 字。

4.8.2 取消接收控制命令

C5 命令是取消接收控制命令。它是与 C4 命令相对应的，当一个接收通道执行了 C4 命令后，可用 C5 命令来取消该通道上的 C4 命令。在执行 C5 命令前,应先在 3FF1 单元处填上接收通道号，范围从 1 到 6。

4.8.3 设置校验位命令

C6 命令是设置校验位命令。在执行该命令前,应先在 3FF1 和 3FF2 单元处填上命令参数,3FF1 单元处填写的是 3282 芯片号,范围从 1 到 3; 3FF2 单元处填写值为 1 或 0，1 代表偶校验，0 代表奇校验。初始值为奇校验。

4.8.4 设置发送器速率命令

C7 命令是设置发送器速率命令。在执行该命令前,应先在 3FF1 和 3FF2 单元处填上命令参数，3FF1 单元处填写的是 3282 芯片号,范围从 1 到 3; 3FF2 单元处填写值为 1 或 0，为 0 时发送器的速率为时钟速率的十分之一，为 1 时发送器的速率为时钟速率的八十分之一。初始值为时钟速率的十分之一。

4.8.5 设置接收器速率命令

C8 命令是设置接收器速率命令。在执行该命令前,应先在 3FF1 和 3FF2 单元处填上命令参数，3FF1 单元处填写的是 3282 芯片号,范围从 1 到 3; 3FF2 单元处填写值为 1 或 0，为 1 时接收器的速率为时钟速率的十分之一，为 1 时接收器的速率为时钟速率的八十分之一。初始值为时钟速率的十分之一。

当改变接收或发送速率时,应根据改变速率的快慢相应的改变数据间隔。如：HT-C302 刚一上电时的初始值是发送速率为 100Kbps，数据间隔为 1.25ms，如果把速率变为 12.5Kbps，那么速率就变慢了，数据间隔也要相应增加，否则就会发生发送错误。

4.8.6 设置数据长度命令

C9 命令是设置校验位命令。在执行该命令前,应先在 3FF1 和 3FF2 单元处填上命令参数,3FF1 单元处填写的是 3282 芯片号,范围从 1 到 3; 3FF2 单元处填写值为 1 或 0，为 1 时一个 ARINC 字的长度为 25 位，为 0 一个 ARINC 字的长度为 32 位。初始值为 32 位。

5 编程说明

5.1 WINDOWS 程序说明

5.1.1 动态链接库说明

HT-C302 板卡在 WINDOW 下的动态链接库文件为 LibC302.dll、LibC302.lib，在其中包含了以下 14 个函数。

a. 发送函数

void Send_C302(HANDLE hndFile,UCHAR Channel,ULONG *SendBuffer,UCHAR Number,UCHAR SendStyle)

功能说明：向双口存储器中的一个发送通道循环发或单发一帧数据。

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

Channel：发送通道号，取值范围为 1、2、3。

Sendbuffer:发送数据缓冲区指针。

Number:发送 ARINC429 字个数。

Sendtyle:发送类型。0 为单发，1 为循环发。

返回值：无

例：Send_C302(hndFile,1,sendbuf,16,1)

说明：向发送通道 1 循环发 16 个数据，发送数据缓冲区指针为 sendbuf。

b. 接收函数

void Receive_C302(HANDLE hndFile,UCHAR Channel,ULONG *ReceBuffer,UCHAR ReceNum)

功能说明：从特定接收通道读出 LABLE 值 00 到 recenum 的数据。

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

Channel：接收通道号，取值范围为 1～6。

Recebuf:接收数据缓冲区指针。

Recenum:最后读取的 LABLE 值。

返回值：无

例：Receive_C302(hndFile,1,recebuf,10)

说明：从接收通道 1 读取 LABLE 值 00 到 0A 共 11 个数据，接收数据缓冲区指针为 recebuf。

c. 设中断标志函数

void SetINTSign_C302(HANDLE hndFile,UCHAR *INTWord);

功能：设置接收时向 PC 机发中断的 LABLE 值。

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

INTWord：指向向 PC 机发中断的 LABLE 值缓冲区的指针。
这个缓冲区中依次存放了六个接收通道向 PC 机发中断的 LABLE 值。

返回值：无

例：SetINTSign_C302 (hndFile,intwordptr)

说明：设置接收时向 PC 机发中断的 LABLE 值，设置的 LABLE 值缓冲区指针为 intwordptr。

d. 写命令字函数

UCHAR SetCommandWord_C302(HANDLE hndFile,UCHAR cWord,USHORT parameter,UCHAR cmd1,UCHAR cmd2)

功　　能：执行命令字

入口参数：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

cWord：命令字:0 表示 C0 命令，1 表示 C1 命令，2 表示 C2 命令，3 表示 C3 命令, 4 表示 C4 命令, 5 表示 C5 命令, 6 表示 C6 命令, 7 表示 C7 命令, 8 表示 C8 命令, 9 表示 C9 命令。

parameter、cmd1、cmd2：命令参数，与命令字有关。详细定义如下：

cWord 为 0、1、2：parameter 表示要设置的时间，cmd1 和 cmd2 没有意义。

cWord 为 3：命令参数没有意义。

cWord 为 4：cmd1 表示接收通道号，cmd2 表示控制字符，parameter 没有意义。

cWord 为 5：cmd1 表示接收通道号，cmd2 和 parameter 没有意义。

cWord 为 6：cmd1 表示 3282 芯片号，cmd2 为 0 时表示奇校验，为 1 时表示偶校验，parameter 没有意义。

cWord 为 7：cmd1 表示 3282 芯片号，cmd2 为 0 时表示发送速率为时钟的十分之一，为 1 时表示发送速率为时钟的八十分之一，parameter 没有意义。

cWord 为 8：cmd1 表示 3282 芯片号，cmd2 为 0 时表示接收速率为时钟的十分之一，为 1 时表示接收速率为时钟的八十分之一，parameter 没有意义。

cWord 为 9：cmd1 表示 3282 芯片号，cmd2 为 0 时表示数据长度为 32 位，为 1 时表示数据长度为 25 位，parameter 没有意义。

返回值:对 C3 命令，它表示自检标志。对其它命令没有意义。

调用例子 1：(UCHAR)selftestsign=CommandWord(hndFile,0,1200,0,0,0)
说明：设置定时器 0 的时间为 1200μs
调用例子 2：(UCHAR)selftestsign=CommandWord(hndFile,3,0,0,0)
说明：执行自检命令，在 selftestsign 中存储自检状态字。
调用例子 3：(UCHAR)selftestsign=CommandWord(hndFile,4,0,1,2)
说明：接收通道 1 只接收第 10、9 两位为 1、0 的 ARINC 字。
调用例子 4：(UCHAR)selftestsign=CommandWord(hndFile,5,0,1,0)
说明：取消上面执行的命令 4。
调用例子 5：(UCHAR)selftestsign=CommandWord(hndFile,6,0,1,1)
说明：第一片 3282 芯片传输数据时为偶校验。。
调用例子 6：(UCHAR)selftestsign=CommandWord(hndFile,7,0,1,1)
说明：设置发送速率为 CLK/80bps。
调用例子 7：(UCHAR)selftestsign=CommandWord(hndFile,8,0,1,1)
说明：设置接收速率为 CLK/80bps。
调用例子 8：(UCHAR)selftestsign=CommandWord(hndFile,9,0,1,1)
说明：设置数据长度为 25 位。

e. 读中断标志函数

UCHAR ReadINTSign_C302(HANDLE hndFile,UCHAR Channel)

功能：读中断标志并把中断标志清零
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。
Channle：接收通道号，取值范围为 1～6。

返回值:返回中断标志。
例：(UCHAR)intsign=ReadIntsign_C302（hndFile,1）
说明：读出接收通道 1 的中断标志，在读出后把中断标志单元清零。

f. 读接收计数函数

USHORT ReadReceiveNum_C302(HANDLE hndFile,UCHAR Channel)

功能：读接收计数
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。
Channel：接收通道号，取值范围为 1～6。

返回值:返回接收计数值。

例：(USHORT)calculate= ReadReceiveNum_C302（hndFile,1）
说明：读出接收通道 1 的接收计数值。

g. 读 429 板卡标志函数

USHORT Init429_C302 (HANDLE hndFile)

功能：读出 429 板的标志,如为 0x429 则板子正常，否则板卡工作不正常。
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

返回值:429 板的标志。

例：(USHORT) t429sign= Init429_C302 (hndFile)
说明：读出 429 板的标志，看是否为 0x429，如是则板卡正常，否则不正常。

h. 发送停止函数

void StopSend_C302(HANDLE hndFile,UCHAR Channel,UCHAR SendStyle)

功能：停止发送。
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。
Channel：发送通道号，取值范围 1～3。
Sendtyle:发送类型。0 为单发，1 为循环发。

返回值：无
例：StopSend_C302(hndFile,1,1)
说明：停止发送通道 1 循环发送数据。

i. 转换 429 格式函数

void Converse429Format_C302(ULONG *buffer,UCHAR contype)

功能：429 编程格式与总线格式转换。
参数说明：
buffer:指向一个 ARINC429 字的缓冲区。
congtype:转换方式。为 0 时从编程格式转换到总线格式，为 1 时从总线格式转换到编程格式。
返回值:无。

例：Converse429Format_C302(buffer,0)
说明：把 buffer 中的数据从编程格式转换到总线格式。

j. 清除接收缓冲区函数

void Clear429Receive_C302(HANDLE hndFile,UCHAR Channel)

功能：清接收缓冲区。
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。
Channel:接收通道号, 取值范围 1～6。
返回值:无。

例：Clear429Receive_C302(hndFile,1);
说明：清除接收通道 1 缓冲区。

k. 请求信号灯函数

void RequestSem_C302(HANDLE hndFile,USHORT SemNum)

功能：请求信号灯。

参数说明：

hndFile:板卡句柄，由OpenDevice_C302（）函数取得

SemNum:信号灯号。

返回值:无。

例：RequestSem_C302(hndFile,0)

说明：请求0号信号灯。

l. 释放信号灯函数

void ReleaseSem_C302(HANDLE hndFile,USHORT SemNum)

功能：释放信号灯。

参数说明：

hndFile:板卡句柄，由OpenDevice_C302（）函数取得。

SemNum:信号灯号。

返回值:无。

例：ReleaseSem_C302 (hndFile,0)

说明：释放0号信号灯。

m. 取得句柄

HANDLE OpenDevice_C302(DWORD instance)

功　　能：取得指向 429 板卡的句柄。执行所有函数前必须首先取得该句柄。

参数说明：

instance:板卡序号，0 表示第一块，以此类推。

返 回 值：指向 429 板卡的句柄。

例：HANDLE hndFile=OpenDevice_C302(0)

说明：取得429板卡句柄hndFile。

n. 释放句柄

void CloseDevice_C302(HANDLE hndFile)

功　　能：释放指向 429 板卡的句柄。退出程序前应释放句柄。

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

返 回 值：无。

例：CloseDevice_C302(hndFile)

说明：释放429板卡句柄hndFile。

5.1.2 演示程序说明

5.1.2.1 演示程序安装

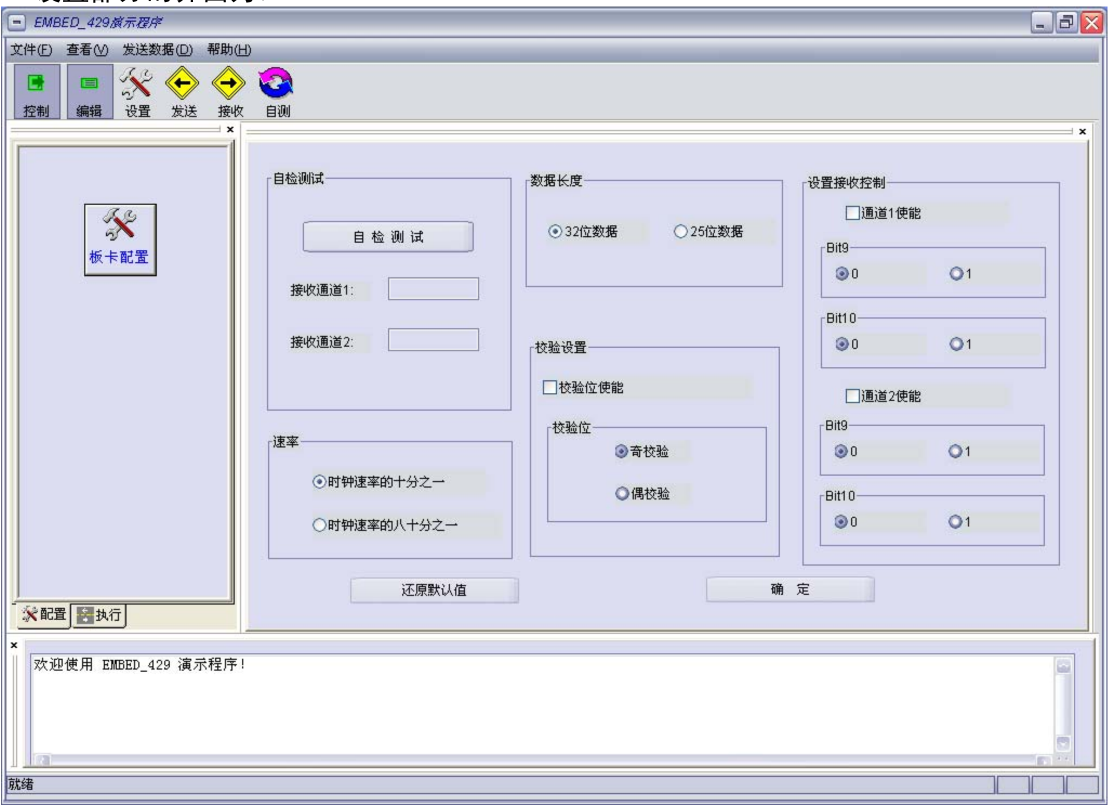
运行配套光盘，选择相应板卡的演示程序的安装即可。

5.1.2.2 演示程序使用说明

演示程序的执行文件为 DemoC302.exe，程序一共分为四个部分：设置、发送、接收和自测。

1） 设置

设置部分的界面为：



2） 发送

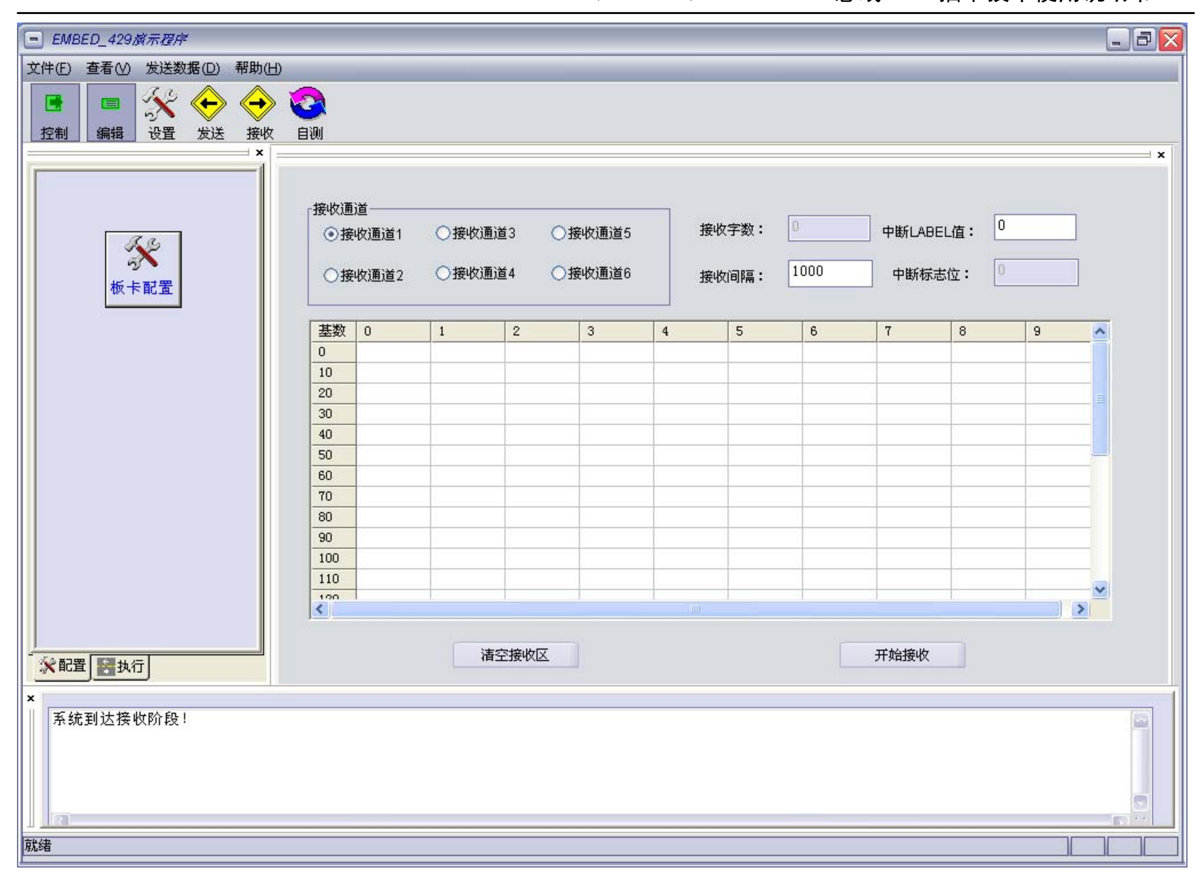
发送部分的界面为：



在发送界面中，可以修改发送通道、发送方式、发送间隔，发送的数据内容可以通过填表填入，也可以通过载入数据的方法载入，表中填写的数据也可以保存下来。

3） 接收

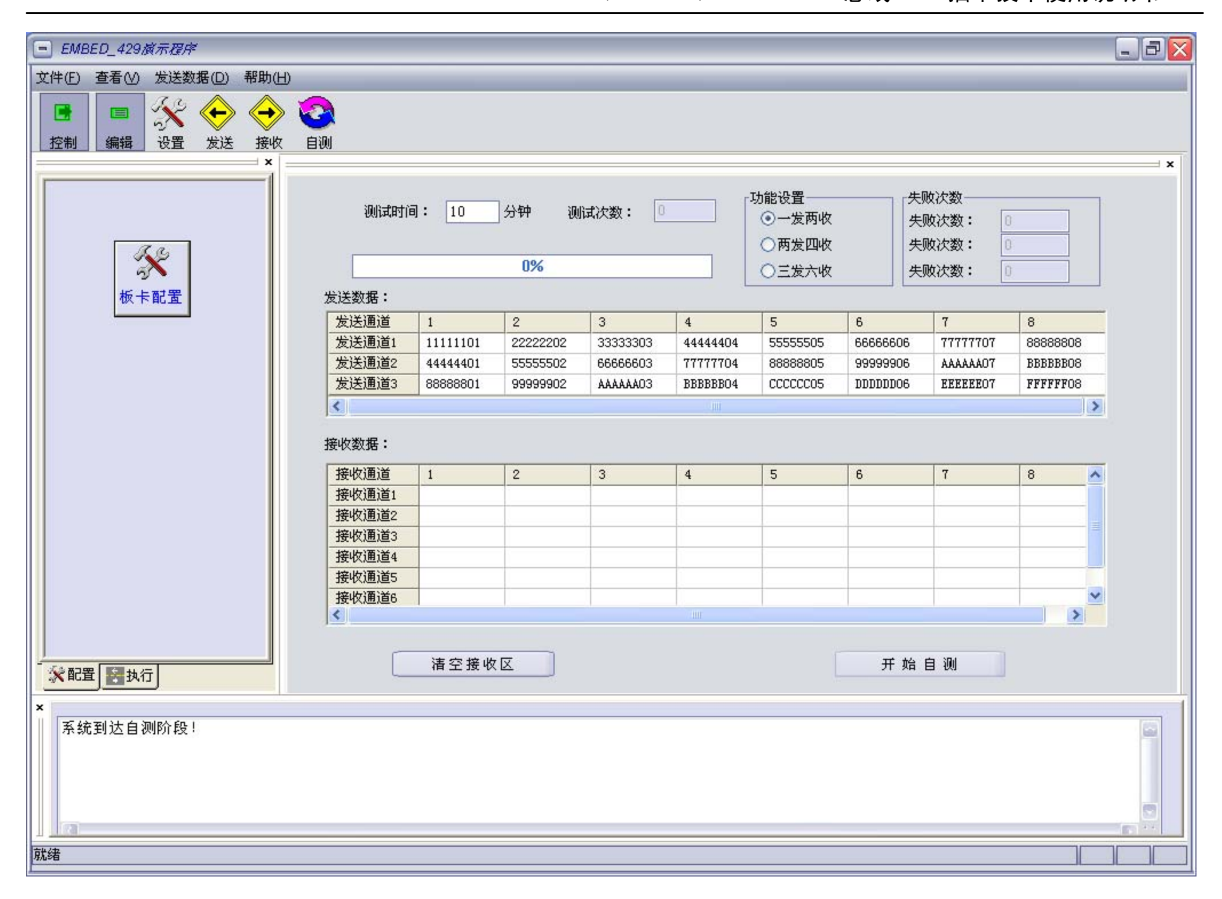
接收部分的界面的为：



在接收界面中，可以选择接收通道和接收间隔，可以周期性的查询接收缓冲区中的内容。

4） 自检

自检部分的界面为：



把板卡的第 1 发送通道与 1、2 接收通道，第 2 发送通道与 3、4 接收通道，第 3 发送通道与 5、6 接收通道相短接，然后自检测试，程序会自动的判断收发是否正常，并把错误计数显示出来。

5.1.3 驱动程序的安装

运行配套光盘，选择相应板卡的驱动安装即可。

5.2 WINCE 下程序说明

5.2.1 驱动程序的安装

WINCE 下的驱动程序与 Windows2000 和 Windows XP 系统下的驱动程序不同，它是以用户态下的动态链接库的形式存在的，通过 WINCE 里的 CEC 文件工具可以把驱动打包变成第三方组件的形式供系统加载。

驱动安装按照以下步骤：

1）首先安装 Platform Builder，安装时最好只改变它的盘符而不要改变它的相对路径，因为系统生成时有时按默认路径拷贝文件。

2）安装完成后在盘符根目录下会有 WINCE420(WINCE500)目录，如:d:\WINCE420(WINCE500)。将光盘中 WinCE 目录下的 Driver-HTC302 目录拷贝到 WINCE420(WINCE500)\PUBLIC 目录下。

3）执行文件 DrvC302-CE.msi，这里将向组件文件夹里添加 DrvC302.cec 文件，并将组件导入系统的 Catalog Features 中，其中完成了注册表修改的操作，如果从 Catalog 列表 Third Party\Device Drivers 中没有看到 HT-C302 Driver,刷新一下列表显示即可。

4）建立目标平台工程后，将 Catalog 中显示的 HT-C302 Driver 加入平台工程，执行 sysgen 编译生成 NK.BIN 镜像文件即可。

5.2.2 动态链接库说明

HT-C302 板卡在 WINDOW CE 下的动态链接库文件为 LibC302.dll、LibC302.lib，在其中包含了以下 14 个函数。

a. 发送函数

__declspec(dllexport) void Send_C302(HANDLE hndFile,UCHAR Channel,ULONG *SendBuffer,UCHAR Number,UCHAR SendStyle)

功能说明：向双口存储器中的一个发送通道循环发或单发一帧数据。

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

Channel：发送通道号，取值范围为 1、2、3。

Sendbuffer:发送数据缓冲区指针。

Number:发送 ARINC429 字个数。

Sendtyle:发送类型。0 为单发，1 为循环发。

返回值：无

例：Send_C302(hndFile,1,sendbuf,16,1)

说明：向发送通道 1 循环发 16 个数据，发送数据缓冲区指针为 sendbuf。

b. 接收函数

__declspec(dllexport) void Receive_C302(HANDLE hndFile,UCHAR Channel,ULONG *ReceBuffer, WORD offlbl,UCHAR ReceNum)

功能说明：从特定接收通道读出 LABLE 值 00 到 recenum 的数据。

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

Channel：接收通道号，取值范围为 1～6。

Recebuf:接收数据缓冲区指针。

Offlbl: 开始读取数据的 lable 值。

Recenum:读取 ARINC 字的个数。

返回值：无

例：Receive_C302(hndFile,1,recebuf,0,10)

说明：从接收通道 1 读取 LABLE 值 00 到 0A 共 11 个数据，接收数据缓冲区指针为 recebuf。

c. 设中断标志函数

__declspec(dllexport) void SetINTSign_C302(HANDLE hndFile,UCHAR *INTWord);

功能：设置接收时向 PC 机发中断的 LABLE 值。

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

INTWord：指向向 PC 机发中断的 LABLE 值缓冲区的指针。
这个缓冲区中依次存放了六个接收通道向 PC 机发中断的 LABLE 值。

返回值：无

例：SetINTSign_C302 (hndFile,intwordptr)

说明：设置接收时向 PC 机发中断的 LABLE 值，设置的 LABLE 值缓冲区指针为 intwordptr。

d. 写命令字函数

__declspec(dllexport) UCHAR SetCommandWord_C302(HANDLE hndFile,UCHAR cWord,USHORT

parameter,UCHAR cmd1,UCHAR cmd2)

功　　能：执行命令字

入口参数：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

cWord：命令字:0 表示 C0 命令，1 表示 C1 命令，2 表示 C2 命令，3 表示 C3 命令, 4 表示 C4 命令, 5 表示 C5 命令, 6 表示 C6 命令, 7 表示 C7 命令, 8 表示 C8 命令, 9 表示 C9 命令。

parameter、cmd1、cmd2：命令参数，与命令字有关。详细定义如下：

cWord 为 0、1、2：parameter 表示要设置的时间，cmd1 和 cmd2 没有意义。

cWord 为 3：命令参数没有意义。

cWord 为 4：cmd1 表示接收通道号，cmd2 表示控制字符，parameter 没有意义。

cWord 为 5：cmd1 表示接收通道号，cmd2 和 parameter 没有意义。

cWord 为 6：cmd1 表示 3282 芯片号，cmd2 为 0 时表示奇校验，为 1 时表示偶校验，parameter 没有意义。

cWord 为 7：cmd1 表示 3282 芯片号，cmd2 为 0 时表示发送速率为时钟的十分之一，为 1 时表示发送速率为时钟的八十分之一，parameter 没有意义。

cWord 为 8：cmd1 表示 3282 芯片号，cmd2 为 0 时表示接收速率为时钟的十分之一，为 1 时表示接收速率为时钟的八十分之一，parameter 没有意义。

cWord 为 9：cmd1 表示 3282 芯片号，cmd2 为 0 时表示数据长度为 32 位，为 1 时表示数据长度为 25 位，parameter 没有意义。

返回值:对 C3 命令，它表示自检标志。对其它命令没有意义。

调用例子 1：(UCHAR)selftestsign=CommandWord(hndFile,0,1200,0,0,0)
说明：设置定时器 0 的时间为 1200μs
调用例子 2：(UCHAR)selftestsign=CommandWord(hndFile,3,0,0,0)
说明：执行自检命令，在 selftestsign 中存储自检状态字。
调用例子 3：(UCHAR)selftestsign=CommandWord(hndFile,4,0,1,2)
说明：接收通道 1 只接收第 10、9 两位为 1、0 的 ARINC 字。
调用例子 4：(UCHAR)selftestsign=CommandWord(hndFile,5,0,1,0)
说明：取消上面执行的命令 4。
调用例子 5：(UCHAR)selftestsign=CommandWord(hndFile,6,0,1,1)
说明：第一片 3282 芯片传输数据时为偶校验。。
调用例子 6：(UCHAR)selftestsign=CommandWord(hndFile,7,0,1,1)
说明：设置发送速率为 CLK/80bps。
调用例子 7：(UCHAR)selftestsign=CommandWord(hndFile,8,0,1,1)
说明：设置接收速率为 CLK/80bps。
调用例子 8：(UCHAR)selftestsign=CommandWord(hndFile,9,0,1,1)
说明：设置数据长度为 25 位。

e. 读中断标志函数

__declspec(dllexport) UCHAR ReadINTSign_C302(HANDLE hndFile,UCHAR Channel)

功能：读中断标志并把中断标志清零

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

Channle：接收通道号, 取值范围为 1～6。

返回值:返回中断标志。

例：(UCHAR)intsign=ReadIntsign_C302（hndFile,1）

说明：读出接收通道 1 的中断标志，在读出后把中断标志单元清零。

f. 读接收计数函数

__declspec(dllexport) USHORT ReadReceiveNum_C302(HANDLE hndFile,UCHAR Channel)

功能：读接收计数
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。
Channel：接收通道号, 取值范围为 1～6。

返回值:返回接收计数值。

例：(USHORT)calculate= ReadReceiveNum_C302（hndFile,1）
说明：读出接收通道 1 的接收计数值。

g. 读 429 板卡标志函数

__declspec(dllexport) USHORT Init429_C302 (HANDLE hndFile)

功能：读出 429 板的标志,如为 0x429 则板子正常，否则板卡工作不正常。
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

返回值:429 板的标志。

例：(USHORT) t429sign= Init429_C302 (hndFile)
说明：读出 429 板的标志，看是否为 0x429，如是则板卡正常，否则不正常。

h. 发送停止函数

__declspec(dllexport) void StopSend_C302(HANDLE hndFile,UCHAR Channel,UCHAR SendStyle)

功能：停止发送。
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。
Channel：发送通道号，取值范围 1～3。
Sendtyle:发送类型。0 为单发，1 为循环发。

返回值：无
例：StopSend_C302(hndFile,1,1)
说明：停止发送通道 1 循环发送数据。

i. 转换 429 格式函数

__declspec(dllexport) void Converse429Format_C302(ULONG *buffer,UCHAR contype)

功能：429 编程格式与总线格式转换。
参数说明：
buffer:指向一个 ARINC429 字的缓冲区。
congtype:转换方式。为 0 时从编程格式转换到总线格式，为 1 时从总线格式转换到编程格式。
返回值:无。

例：Converse429Format_C302(buffer,0)
说明：把 buffer 中的数据从编程格式转换到总线格式。

j. 清除接收缓冲区函数

__declspec(dllexport) void Clear429Receive_C302(HANDLE hndFile,UCHAR Channel)

功能：清接收缓冲区。
参数说明：
hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。
Channel:接收通道号, 取值范围 1～6。
返回值:无。

例：Clear429Receive_C302(hndFile,1);
说明：清除接收通道 1 缓冲区。

k. 请求信号灯函数

```
__declspec(dllexport) void RequestSem_C302(HANDLE hndFile,USHORT SemNum)
```

功能：请求信号灯。
参数说明：
hndFile:板卡句柄，由OpenDevice_C302（）函数取得
SemNum:信号灯号。
返回值:无。

例：RequestSem_C302(hndFile,0)
说明：请求0号信号灯。

l. 释放信号灯函数

```
__declspec(dllexport) void ReleaseSem_C302(HANDLE hndFile,USHORT SemNum)
```

功能：释放信号灯。
参数说明：
hndFile:板卡句柄，由OpenDevice_C302（）函数取得。
SemNum:信号灯号。
返回值:无。

例：ReleaseSem_C302 (hndFile,0)
说明：释放0号信号灯。

m. 取得句柄

```
__declspec(dllexport) HANDLE OpenDevice_C302(DWORD instance)
```

功　　能：取得指向 429 板卡的句柄。执行所有函数前必须首先取得该句柄。
参数说明：
instance:板卡序号，0 表示第一块，以此类推。
返 回 值：指向 429 板卡的句柄。

例：HANDLE hndFile=OpenDevice_C302(0)
说明：取得429板卡句柄hndFile。

n. 释放句柄

```
__declspec(dllexport) void CloseDevice_C302(HANDLE hndFile)
```

功　　能：释放指向 429 板卡的句柄。退出程序前应释放句柄。

参数说明：

hndFile:板卡句柄，由 OpenDevice_C302（）函数取得。

返 回 值：无。

例：CloseDevice_C302(hndFile)

说明：释放 429 板卡句柄 hndFile。

5.2.3 演示程序说明

演示程序的名称为 PCI_429_DEMO.exe，演示程序不需要安装，只要将其存放在 WinCE 可以访问到的存储介质上，并且与动态链接库 LibC302.dll 存放在相同目录下，双击即可运行。点击菜单栏的<Setting>项中的子项，可以相应地选择板卡号和设定板卡向主机产生中断的 LABLE 值，点击菜单栏的<Command>项，可以向板卡发送配置命令，点击<Receive/Send>菜单栏中的相应项可以进行数据的收发和停止。

5.3 Linux 下程序说明

5.3 Linux 下程序说明

5.3.1 驱动程序的安装

驱动程序的安装需要以下几个步骤，某些步骤需要超级用户的权限。

1）驱动程序的编译和驱动文件的安装

如果提供给用户的是驱动程的源代码，则首先需要驱动的编译。进入驱动程序所在的目录执行以下命令：

make clean

make

通过编译或其它途径获得驱动驱动目标文件后，便可以通过以下命令进行驱动文件的安装：

make install

2）创建设备文件

还是在驱动程序目录中，执行以下命令：

./mknode.sh

设备文件的默认主设备号是 200。如果想使用其它的主设备号，则需要执行如下带参数的命令：

./mknode.sh –m MAJOR_NUM

此处‘MAJOR_NUM‘代表用户指定的主设备号。

3）驱动模块的加载和卸载

在驱动可用前需要将驱动模块加载到内核，可以通过以下命令实现：

insmod arinc429

如果用户需要指定主设备号，则需要在命令中使用选项 major= MAJOR_NUM，‘MAJOR_NUM‘是用户指定的主设备号，注意此设备号应和上面给设备文件指定的设备号一致。例如：

insmod arinc429 major=150

如果用户使用的是 ISA 接口的板卡，则还应该使用选项指定板卡的主机双口存贮器和信号灯 Memory 基地址，选项格式为：ioaddr=ADDR1,ADDR2[[,ADDR1,ADDR2]...]，两个地址为一组，最多 4 组。例如：

```
insmod arinc429 major=150 ioaddr=0xd0000,0xd4000
```

则表示指定的基地址为 0xd0000 和 0xd4000.

卸载驱动程序则相对简单，执行如下命令即可：

```
rmmod arinc429
```

4）驱动文件的反安装

如果需要从系统文件目录中卸载驱动文件，则需进入驱动程序所在的原始目录，执行如下命令：

```
make uninstall
```

强烈建议！在删除驱动后，同时删除所建立的设备文件，对这些遗留设备文件的不当使用有可能会伤害其它使用相同主设备号的设备。

5.3.2 函数库的安装

在函数库的安装过程中某些步骤需要超级用户的权限。

1）函数库的编译和库文件的安装

如果用户得到的是函数库的源文件，则首先需要库的编译。进入函数库所在的目录，执行以下命令：

```
make clean
make
```

获得函数库目标文件后，接下来进行函数库文件的安装，执行以下命令：

```
make install
```

2）函数库的反安装

在原始函数库目录中执行以下命令，从系统中移除函数库文件：

```
make uninstall
```

5.3.3 应用程序与函数库的链接

如果应用程序中使用到函数库中的函数，则需要在应用程序编译链接时使用命令选项-l 指定需链接的库 429ctl，例如：

```
gcc -Wall -g –l429ctl myapp.c -o myapp
```

此命令执行后将把应用程序 myapp.c 和库 429ctl 链接起来。

5.3.4 库函数说明

a. extern int OpenDevice_429(int devIndex)

功　能

打开设备，获得进一步操作的设备句柄。

参数说明

@devIndex ：设备实例索引号（0-3）

返回值

1 ：成功

0 ：失败

b. extern int CloseDevice_429(int devIndex)

功 能

关闭设备，释放所获的设备句柄。

参数说明

@devIndex ：设备实例索引号（0-3）

返回值

1 ：成功

0 ：失败

c. extern int Init_429(int devIndex)

功 能

检查设备是否是所支持的设备，然后对其初始化。

参数说明

@devIndex ：设备实例索引号（0-3）

返回值

1 ：成功

0 ：失败

d. extern int RequestSem_429(int devIndex,unsigned short SemNum)

功 能

请求访问双口存储器的信号量。

参数说明

@devIndex ：设备实例索引号（0-3）

@SemNum ：信号量索引号

0 到 5 对应接收通道 1 到 6；

6 对应发送通道；

返回值

1 ：成功

0 ：失败

e. extern int ReleaseSem_429(int devIndex, unsigned short SemNum)

功 能

释放请求的双口存储器访问信号量。

参数说明

@devIndex ： 设备实例索引号（0-3）

@SemNum ： 信号量索引号

0 到 5 对应接收通道 1 到 6；

6 对应发送通道

返回值

1 ： 成功

0 ： 失败

f. extern int Clear429Receive_429(int devIndex, unsigned char Channel)

功 能

清空板内接收缓冲区。

参数说明

@devIndex ： 设备实例索引号（0-3）

@Channel ： 通道号（1-6）

返回值

1 ： 成功

0 ： 失败

g. extern int SetINTSign_429(int devIndex, unsigned char *INTWord)

功 能

设置产生主机中断的 arinc 字的 LABLE 值。

参数说明

@devIndex ： 设备实例索引号（0-3）

@INTWord ： 指向需设置的 LABLE 的缓冲区

返回值

1 ： 成功

0 ： 失败

h. extern unsigned char ReadINTSign_429(int devIndex, unsigned char Channel)

功 能

读中断标志并把中断标志清零。

参数说明

@devIndex ： 设备实例索引号（0-3）

@Channel ： 接收通道号（1-6）

返回值

中断标志

i. extern unsigned char SetCommandWord_429(int devIndex,
```
                                            unsigned char cWord,
                                            unsigned short parameter,
                                            unsigned char cmd1,
                                            unsigned char cmd2 )
```

功　能

执行命令字。

参数说明

@devIndex　：设备实例索引号（0-3）

@cWord　：命令字（0 到 9 对应命令字 C0 到 C9）

@parameter　：参见 Windows 库函数说明

@cmd1　：参见 Windows 库函数说明

@cmd2　：参见 Windows 库函数说明

返回值

对 C3 命令为自检标志；

对其它命令没有意义。

j. extern int Send_429(int devIndex, unsigned char Channel,
```
                        unsigned long *SendBuffer, unsigned char Number,
                        unsigned char SendStyle )
```

功　能

发送数据。

参数说明

@devIndex　：设备实例索引号（0-3）

@Channel　：发送通道号（1-3）

@SendBuffer：发送缓冲区

@Number　：要发送的 ARINC 字个数

@SendStyle　：发送方式（0：单次发，1：循环发）

返回值

1　：成功

0　：失败

k. extern int StopSend_429(int devIndex, unsigned char Channel,
```
                                            unsigned char SendStyle)
```

功　能

停止发送。

参数说明

@devIndex　：设备实例索引号（0-3）

@Channel　：发送通道号，1 到 3 对应通道 1 到 3

@SendStyle ： 发送方式。（0：单次发，1：循环发）

返回值

1 ： 成功

0 ： 失败

l. extern unsigned short ReadReceiveNum_429(int devIndex, unsigned char Channel)

功 能

读接收计数。

参数说明

@devIndex ： 设备实例索引号（0-3）

@Channel ： 接收通道号（1-6）

返回值

通道接收计数值

m. extern int Receive_429(int devIndex, unsigned char Channel,
unsigned char label, unsigned long *ReceBuffer,
unsigned char ReceNum)

功 能

数据接收函数。

参数说明

@devIndex ： 设备实例索引号（0-3）

@Channel ： 通道号（1-6）

@ReceBuffer ： 接收缓冲区指针

@ReceNum ： 最后读取的 LABLE 值

返回值

1 ： 成功

0 ： 失败

5.4 VxWorks 下程序说明

5.4.1 驱动程序的安装

VxWorks 下的驱动安装需要按照下列步骤进行：

1） 在 Tornado 环境中新建工程，并将光盘中带的 DrvC302.o 文件拷贝到新建工程的目录下。

2） 点击新建工程空间中的 Builds 选项卡，双击 default 项，选择 Macros 选项卡，然后从该选项卡的 Macros 下拉列表中选择 EXTRA_MODULES 项，在其 Value 域中填入 $(PRJ_DIR)/DrvC302.o，点击左下角的<Add/Set>按钮，选择 OK 完成设置。

3） 编译生成 VxWorks 镜像文件，驱动安装完成。

5.4.2 驱动函数说明

VxWorks 下的驱动函数全部采用标准系统调用的方式实现，用户在编程时可以采用 open、close、read、write、ioctl 等系统调用实现对设备的控制，下面对各个系统调用在本卡驱动中实现的功能进行说明。

a. 驱动初始化函数：

函数定义：STATUS HTC302Drv(void);

功　　能：初始化驱动程序数据结构，安装驱动程序处理函数，读板卡标志，并创建设备节点。

参数说明：板卡基地址值，根据板卡实际跳线设置

中断号，根据板卡跳线设置，该参数设为 0 时表示不采用中断方式

返 回 值：OK 初始化成功，成功后将在系统中创建设备节点，节点名称定义如下：

/HTC302/0　　接收通道 1
/HTC302/1　　接收通道 2
/HTC302/2　　接收通道 3
/HTC302/3　　接收通道 4
/HTC302/4　　接收通道 5
/HTC302/5　　接收通道 6
/HTC302/6　　发送通道 1
/HTC302/7　　发送通道 2
/HTC302/8　　发送通道 3

ERROR 初始化失败

例　　子：HTC302Drv();

b. 打开设备函数

函数定义：int open(const char * name, int flags, int mode);

功　　能：获得设备句柄

参数说明：设备节点名称字符串

打开方式

操作权限

返 回 值：正确时返回设备句柄的整型值;

错误时返回 ERROR。

例　　子：fd1=open("/HTC302/6",O_RDWR,0644) //打开发送通道 1

c. 关闭设备函数

函数定义：STATUS close(int fd);

功　　能：关闭设备

参数说明：设备节点句柄

返 回 值：正确时返回设备句柄的整型值;

错误时返回 ERROR，表示该设备节点未打开。

例　　子：close(fd1); //关闭打开的设备节点

d. 读操作函数

函数定义：int read(int fd, char * buffer, size_t maxbytes);

功　　能：从接收设备读取数据

参数说明：已打开的设备节点句柄

接收缓冲区，该缓冲区的数据结构如下：

```
typedef struct
{
    char   offlbl;     /* lable to receive from*/
```

```
    ULONG Data[256];  /* data buffer for receiving */
}ARINCRcvMsg;
```

该数据结构在 EXPORT_C302.h 中定义。

Maxbytes 在此处表示从 offlbl 开始接收的 ARINC 字个数。

返 回 值：正确时返回接收的 ARINC 字个数；

错误时返回 ERROR，表示该设备节点未打开。

例　　子：

```
#include "EXPORT_C302.h"
  ARINCRcvMsg RcvMsg0;
  RcvMsg0.offlbl=0;   /*receive data from label 0*/
  read(fd2,(char *)&RcvMsg0,rcvnum);
```

e. 写操作函数

函数定义：int write(int fd, char * buffer, size_t nbytes);

功　　能：向发送设备写入数据

参数说明：已打开的设备节点句柄

发送缓冲区，该缓冲区的数据结构如下：

```
typedef struct
{
    char    SendMode; /* Send Mode: 0--single send; 1--loop send */
    ULONG Data[256];  /* data buffer for sending */
} ARINCSendMsg;
```

该数据结构在 EXPORT_C302.h 中定义。

nbytes 在此处表示要发送的 ARINC 字个数。

返 回 值：正确时返回接收的 ARINC 字个数；

错误时返回 ERROR，表示该设备节点未打开。

例　　子：

```
#include "EXPORT_C302.h"
ARINCSendMsg SendMsg;
SendMsg.SendMode=0;
write(fd1,(char *)&SendMsg,sendnum);
```

f. ioctl 操作函数

函数定义：int ioctl(int fd, int function, int arg);

功　　能：实现对设备的基本配置操作

参数说明：fd，已打开的设备节点句柄

function，要进行的配置操作命令

arg，提供给配置操作的参数

具体的命令及参数请参见下表。

function	arg	功能说明	返回值
0x11	无意义	清空双口 RAM 该通道接收缓冲区中的数据，并将接收计数归零	正确返回 OK。错误返回 ERROR，表示设备句柄错误。
0x13	无意义	读中断标志，并将其清零	正确返回该通道的中断标志；错误返回 ERROR，表示设备句柄错误。
0x14	0 或 1	停止发送，参数值：0 表示单发方式；1 表示循环发送方式	正确返回 OK；错误返回 ERROR，表示设备句柄错误。
0x15	无意义	读接收计数	正确返回该通道的接收计数值；错误返回 ERROR，表示设备句柄错误。
0x16	Lable	设置接收时向 PC 机发送发中断的 LABLE 值	正确返回 OK；错误返回 ERROR，表示设备句柄错误。
0x17	无意义	读命令字缓冲区的状态，为 0 时表示上一次的命令已经执行完	返回 OK 时表示上一次命令执行完成，可以继续进行命令缓冲区操作，

		成，否则未完成	返回 ERROR 时未完成，需等待
0xc0	数据间隔（μs）	设置发送通道 1 的发送数据间隔	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令
0xc1	数据间隔（μs）	设置发送通道 2 的发送数据间隔	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令
0xc2	数据间隔（μs）	设置发送通道 3 的发送数据间隔	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令
0xc3	无意义	进行板卡自检测试	执行完成时返回板卡自检标志，返回 ERROR 表示命令缓冲区有未完成的命令
0xc4	通道号和控制字	设置接收控制命令，参数值：4～7 位表示通道号，范围 1～6。0～3 位表示接收控制字，最低两位有效	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令或通道号错误
0xc5	通道号	取消接收控制命令，参数值：4～7 位表示通道号，范围 1～6。0～3 位无效	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令或通道号错误
0xc6	芯片号和校验方式	设置校验位命令，参数值：4～7 位表示芯片号，范围 1～3。0～3 位表示校验方式，0 为奇校验，1 为偶校验	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令或参数错误
0xc7	芯片号和发送速率	设置发送器速率命令，参数值：4～7 位表示芯片号，范围 1～3。0～3 位表示发送速率，0 表示发送器速率为时钟速率的十分之一，1 表示发送器速率为时钟速率的八分之一	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令或参数错误
0xc8	芯片号和接收速率	设置接收器速率命令，参数值：4～7 位表示芯片号，范围 1～3。0～3 位表示接收速率，0 表示接收器速率为时钟速率的十分之一，1 表示接收器速率为时钟速率的八分之一	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令或参数错误
0xc9	芯片号和数据长度	设置数据长度命令，参数值：4～7 位表示芯片号，范围 1～3。0～3 位表示数据长度，0 表示一个 ARINC 字的长度是 32 位，1 表示一个 ARINC 字的长度是 25 位	返回 OK 表示执行完成；返回 ERROR 表示命令缓冲区有未完成的命令或参数错误

5.4.3 演示程序说明

要使用板卡驱动程序，首先要在应用程序的开始部分调用驱动程序中的函数 HTC302Drv 来初始化板卡，例如：HTC302Drv()。演示程序 testC302.c 演示了如何在 VxWorks 下通过标准系统调用函数操作 HTC302 板卡，要开始演示程序，可将 testC302.o 下载到目标机中，也可以将 testC302.c 复制到 BSP 目录中，连同系统一起编译。在目标 shell 中输入 HTC302()，或在启动代码中直接调用 HTC302()函数来测试板卡，程序首先进行单发测试，发送通道 1 发送数据，接收通道 1 和 2 接收数据；然后进行循环发送测试，其中演示了各个系统调用的使用方式。

6 装箱清单

HT-C302 板	1 块
随机光盘	1 片
测试电缆	1 根